TITLE

MEMORY UNIT AND MEMORY MODULE USING THE SAME

BACKGROUND OF THE INVENTION

Field of the Invention

5   The present invention relates to cache memory, and more particularly, the invention relates to a memory unit and a memory module applicable to a cache memory.

Description of the Related Art

Cache memory is located functionally between a processor and a main memory. The cache memory is generally faster but smaller than the main memory. The cache memory decreases memory access time by storing copies of portions of the contents of main memory. When a local process performs certain "cachable" operations requiring memory access, the cache memory is checked first to determine if it contains a copy of the information to be accessed. If the cache memory contains a copy of the information to be accessed, the processor performs the operation using the cache memory without accessing main memory. A cache "hit" occurs when the information to be accessed by a cachable operation is stored in the cache memory, and a cache "miss" occurs when the information is not.

A main memory address consists of two fields (not shown). The higher order bits are called the tag field, and the remaining lower order bits are called the index field. As shown in Fig. 1, the cache memory 10 uses a tag memory, separate from data memory, to store the tag fields of the addresses of the main memory contents presently stored. When the

1

processor performs memory access, the index field of the address addresses the tag memory. The output data from the tag memory is then compared bit by bit with the tag field of the main memory address. If the two fields are identical, a hit has occurred, and the corresponding data is supplied to the processor. If the fields are not identical, it is a miss, and main memory must be accessed.

If the contents of the tag memory become corrupted, memory accesses that should miss actually hit. This results in incorrect data supplied to the processor, a dangerous situation. It is common to use a signal parity bit over each byte in the tag memory, whereas, if a bad parity is detected when the tag memory is read, a miss condition is forced.

Conventionally, all information in tag memory is stored by SRAM cells, each having structure as shown in Fig.2a. During flushing operation, the tag memory is invalidated entry by entry and requires sequential circuits such that the performance of the cache system is affected.

Another conventional method stores the parity bit (control field) of the tag memory using memory banks with new cell structures as shown in Fig. 2b and 2c, and conventional SRAM cells as shown in Fig. 2a store information of the address field, attribute field and so on. Virtual ground/power terminals (VG/VP) of all cells for the parity bit in the tag memory are connected together to be forced ground/power voltage for flush operations. Although this method requires fewer additional control logic circuits, it requires asynchronous circuits and is poorly compatible with timing analysis. Further, this method has no tolerance for that there cannot be any signal glitch or system noise between the

virtual power/ground terminals and the memory units, with which the data stored in the memory units may be unreliable.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to .5 provide a memory unit with a simple structure to store the parity bit (control field) of the tag memory.

Another objection of the present invention is to provide a memory unit acting as a real synchronous element and is less sensitive to the signal glitch and system noise.

10 According to the above mentioned object, the present invention provides a memory unit. In the memory unit, a first access transistor and a second access transistor are coupled to a bit line pair. The first access transistor has a first terminal coupled to one bit line pair, and the second access 15 transistor has a first terminal coupled to the other. A latch node is coupled between second terminals of the first access transistor and the second access transistor to latch data. A selection unit has two input terminals coupled to a word line and a flush line, and an output terminal is coupled to 20 gates of the first access transistor and the second assess transistor. Invalidation information is written into the latch node from the bit line pair according to activations of the word line or the flush line.

The present invention also provides a memory module. In 25 the memory module, at least one first memory region has a plurality of memory units. In each memory unit, the first access transistor has a first terminal coupled to one of a bit line pair, and the second access transistor has a first terminal coupled to the other. A latch node is coupled between

second terminals of the first access transistor and the second

access transistor to latch data. A selection unit has a first

input terminal coupled to a word line, an output terminal

coupled to gates of the first access transistor and the second

5   assess transistor, and a second input terminal. The second

input terminals of the selection units in all memory units

are coupled to a flush line. Invalidation information is

written into the latch nodes in the memory units from the bit

line pair when the flush line is activated during a flush.

10   **BRIEF DESCRIPTION OF THE DRAWINGS**

The present invention can be more fully understood by

the subsequent detailed description and examples with

reference made to the accompanying drawings, wherein:

Fig. 1 is a diagram of a cache memory;

15   Fig. 2a shows a conventional memory unit;

Fig. 2b shows another conventional memory unit;

Fig. 2c shows another conventional memory unit;

Fig. 3 is a diagram of the memory module according to

the present invention;

20   Fig. 4 is a diagram of the memory units in the memory

module according to the present invention;

Fig. 5 shows the structure of the memory unit according

to the present invention; and

Fig. 6 is a flowchart of the fabrication procedure of

25   a cache memory according to the present invention.

                    DETAILED DESCRIPTION OF THE INVENTION

Fig. 3 is a diagram of the memory module 20 according

to the present invention. The memory module 20 includes two

memory regions M1 and M2, a driving buffer 22, and a D-type Flip-Flop 24. The memory region M1 has a plurality of memory units storing the address field and attribute field in a tag memory. In the memory region M1, all memory units are convention SRAM cells as shown in Fig. 2a. The memory region M2 also includes a plurality of memory units storing entry control bits of the control field in a tag memory. The entry control bits also can be tag validation bits in a cache memory, tag modified/dirty bit in a copy-back/write-back cache memory and the like. Fig. 4 is a diagram of the memory region M2. In the memory region M2, the memory unit MU1 is coupled to the word line WL1, the memory unit MU2 is coupled to the word line WL2 and so on. Further, all memory units MU1~MUn are coupled to a bit line pair (BL and $\overline{BL}$), and a flush line FC. The D-type Flip-Flop (DFF) 24 synchronizes a flush signal FCS by a clock signal CLK. The driver buffer 22 is coupled to the DFF 24 to activate the flush line when receiving the flush signal FCS from the DFF 24. In the embodiment, the driving buffer 22 can include buffers connected in series (not shown).

The structure of each memory unit (MU1~MUn) is shown in Fig. 5. As shown in Fig. 5, each memory unit has a first access transistor T1, a second access transistor T2, a latch node L1 and a selection unit OR1. In the present invention, the selection unit OR1 can be an OR gate. The selection unit OR1 has two input terminals coupled to a word line WL and the flush line FC, and an output terminal coupled to the gates of the first access transistor T1 and the second access transistor T2. The first access transistor T1 has a first terminal coupled to the bit line $\overline{BL}$, and a second terminal coupled to one end of the latch node L1. The second access transistor

5

T2 has a first terminal coupled to the bit line BL and a second terminal coupled to the other end of the latch node L1. In the present invention, the latch node L1 has a first inverter INV1 and a second inverter INV2. The inverter INV1 has an input terminal coupled to the second terminals of the second access transistors T2, and an output terminal coupled to the second terminal of the first access transistors T1. The inverter INV2 has an input terminal coupled to the second terminals of the first access transistors T1, and an output terminal coupled to the second terminal of the second access transistors T2.

During normal operation, the flush line FC is inactive and invalidation information can be written into the latch nodes L1 in the memory unit MU1~MUn from the bit line pair (BL and $\overline{BL}$) according to activation of the corresponding word lines, and the memory unit acts as a conventional memory unit as shown in Fig. 1a. For example, when the word line WL1 is activated, the access transistors T1 and T2 are both turned on such that invalidation information is written to the latch node L1 in the memory unit MU1 from the bit line pair (BL and $\overline{BL}$). Also, when the word line WL2 is activated, the access transistors T1 and T2 are both turned on such that invalidation information is written to the latch node L1 in the memory unit MU2 from the bit line pair (BL and $\overline{BL}$), and so on.

During a flush, all latch nodes L1 in the memory unit MU1~MUn need to be invalidated at once regardless of activation of the word lines WL1~WLn. At this time, the driving buffer actives the flush line FC by the flush control signal FCS from the DFF 24, such that invalidation information is written to the latch nodes L1 in the memory unit MU1~MUn

from the bit line pair (BL and $\overline{BL}$) when the flush line FC is activated.

A conventional method uses memory units, as shown in Fig. 2b and Fig. 2c, to store entry control bits in cache memory, and to change the state of the latch node by controlling the virtual ground/power connected to the source/drain side when the word lines are inactive during flush. Thus, in this conventional method, the memory units are asynchronous elements, and are sensitive to signal glitch, system noise, manufacture process and the like. In the present invention, during flush operation, however, the state of the latch nodes is changed when the word lines and flush lines are activated, such that the memory units in the present invention are real synchronous elements and less sensitive to signal glitch, system noise, manufacture process and so on. Further, the present invention is well suitable to timing analysis, since the state of the latch nodes is changed by controlling the gate side of the access transistor without drain/source side. Moreover, in the present invention, the memory module 20 is applicable to direct-mapping storage, set-associative mapping storage and full-associative mapping storage.

The present invention further provides a fabricating procedure for a cache memory, as shown in Fig. 6. This fabricating procedure for a cache memory is described as follows.

First, in step S100, a specification for a desired cache memory is determined. For example, the specification may include entry number requirement, speed requirement, performance requirement and the like. Next, in step S200, conditions of a tag memory in the desired cache memory is

determined. In this case, the conditions include at least the size and flush cycle time of the tag memory and the like. In step 300, the tag memory is then implemented as an IP module or an object according to the conditions determined in step

5   S200. The tag memory implemented to store entry control bits of the control field in a tag memory uses a plurality of memory units with structure as shown in Fig. 5. The entry control bit can also includes tag validation bits in a cache memory, tag modified/dirty bit in a copy-back/write-back cache memory

10   and the like. The implemented tag memory further stores the address field and attributed field using memory units with the structure shown in Fig. 2a.

Next, in step S400, the implemented tag memory is simulated to check for the conditions determined in step S200.

15   If not, the procedure returns to step S200.

Step S500 is executed if the implemented tag memory meets the conditions determined in step S200. In Step S500, peripheral elements in the cache memory are implemented according to the specification for the desired cache memory.

20   For example, a D-type Flip-Flop and a driving buffer are implemented in the cache memory to provide a flush control signal to the tag memory for flush. In Step S600, the peripheral elements are then modified to match the tag memory, such that the peripheral elements meet the desired

25   specification. For example, a D-type Flip-Flop and a driving buffer are modified repetitively to provide a flush control signal with the requisite time cycle. Finally, step S700 is executed if the peripheral elements meet the desired specification. In step S700, integration of the tag memory

30   and the peripheral elements in the cache memory is simulated

to ensure that integration of the tag memory and the peripheral elements meets the desired specification. If not, this procedure returns to step S200. If so, the cache memory is complete with the desired specification.

5   While the invention has been described by way of example and in terms of the preferred embodiments, it is to be understood that the invention is not limited to the disclosed embodiments. To the contrary, it is intended to cover various modifications and similar arrangements (as would be apparent

10   to those skilled in the art). Therefore, the scope of the appended claims should be accorded the broadest interpretation so as to encompass all such modifications and similar arrangements.